In the Claims

The status of claims in the case is as follows:

1. [Currently amended] A method for control and management of communication traffic, comprising the steps of:

expressing access rules as filters referencing system kernel data;

for outbound processing, determining source application indicia;

for inbound packet processing, executing a look-ahead function to determine target application indicia; said look-ahead function being executed within an IP layer of a protocol stack including ~~an IP layer~~ said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and

responsive to said source or target application indicia, executing filter processing; said filter processing including constructing and evaluating logical expressions including non-IP packet attributes of arbitrary length, and selectively using a set of logical operators, alternative filter selector fields, and value set.

2. [Previously presented] The method of claim 1, wherein said protocol stack is a TCP/IP protocol stack, and further comprising the steps of executing said determining and executing steps within a kernel filtering function upon encountering a filter selector field referencing kernel data not included in said packet.

3. [Previously presented] The method of claim 1, wherein said protocol stack is a TCP/IP protocol stack, and said filter processing including the steps of:

determining a task or thread identifier;

based on said task or thread identifier, determining a process or job identifier; and

based on said process or job identifier, determining job or process attributes for filter processing.

4. [Previously presented] The method of claim 1, wherein said protocol stack is a TCP/IP protocol stack, and said filter processing including the steps of:

determining a user identifier; and

based on said user identifier, determining user attributes for filter processing.

5. [Original] The method of claim 3, further comprising the step of determining from said task identifier a work control block containing said process or job identifier.

6. [Canceled]

7. [Canceled]

8. [Previously presented] The method of claim 1, wherein said protocol stack is a TCP/IP protocol stack, and further comprising the steps of:

delivering to said filters infrastructure access rules for defining security context.

9. [Original] The method of claim 8, said infrastructure including logging, auditing, and filter rule load controls.

10. [Currently amended] A method for control and management of aspects of communication traffic within filtering, comprising the steps of:

receiving IP packet data into a TCP/IP protocol stack executing within a system kernel;

for an inbound IP packet, executing a look-ahead function within an IP layer of a protocol stack including ~~an IP layer~~ said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and

executing filtering code within said IP layer of said system kernel with respect to non-IP packet data accessed within said system kernel outside of said TCP/IP protocol stack; said filtering code constructing and evaluating logical expressions of arbitrary length, and selectively using a set of logical operators, alternative filter selector fields, and value set.

11. [Original] The method of claim 10, said non-IP packet data including context data regarding said IP packet.

12. [Original] The method of claim 10, said non-IP packet data including data specific to a task generating said non-IP packet data.

13. [Original] The method of claim 10, said non-IP packet data including data specific to a task that will receive said IP packet.

14. [Original] The method of claim 11, said context data including packet arrival interface indicia.

15. [Canceled]

16. [Canceled]

17. [Canceled]

18. [Currently amended] A method for centralizing system-wide communication management and control within filter rules, comprising the steps of:

providing filter statements syntax for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values;

for an inbound packet, executing a look-ahead function within an IP layer of a protocol stack including ~~an IP~~ said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered by said sockets layer;

said selector referencing data that does not exist in IP packets;

processing said filter statements, including constructing and evaluating logical expressions of arbitrary length including non-IP packet attributes, and selectively using a set of logical operators, alternative filter selector fields, and value set.

19. [Previously presented] The method of claim 18, wherein said protocol stack is a TCP/IP protocol stack, and said parameters selectively including userid, user profile, user class, user group, user group authority, user special authority, job name, process name, job group, job class, job priority, other job or process attributes, and date & time.

20. [Previously presented] The method of claim 18, wherein said protocol stack is a TCP/IP protocol stack, and said filters statements being provided within a user interface to said system.

21. [Previously presented] The method of claim 18, wherein said protocol stack is a TCP/IP protocol stack, and further comprising the steps of:

establishing a tunnel between two IP address limiting traffic to applications bound to ports at each end of said tunnel;

said filtering code accessing filtering attributes further limiting traffic selectively to job indicia; and

operating said filtering code within a kernel filtering function upon encountering a filter selector field referencing kernel data not included in said traffic.

22. [Currently amended] A method for traversing a portion only of a protocol stack to disallow selective IP packet traffic, comprising the steps of:

receiving a packet in the system kernel of the operating system of a first node from an application, said kernel including a filter processor; said filter processor for constructing and evaluating logical expressions of arbitrary length including non-IP packet attributes, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set;

for inbound packet processing to a first node from a second node, executing a look-ahead function in an IP layer of ~~the system~~ said system kernel of said first node to determine a target application; said system kernel including a TCP/IP protocol stack including ~~an IP layer~~ said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia identifying the application layer application to which said packet would have been delivered;

for both said inbound packet processing, and for outbound packet processing from said first node to said second node, executing within said kernel the steps of processing said packet by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

determining a user ID, process or job identifier from said work control block;

from the user ID, process or job identifier selectively determining attributes for said user process or job; and

passing said attributes to said filter processor for managing and controlling communication traffic.

23. [Currently amended] A method for expressing access rules as filters, comprising the steps of:

providing a filter statements syntax for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values; and said selector referencing data that does not exist in IP packets for controlling access to an application;

for an inbound IP packet, executing a look-ahead function within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and

processing said filter statements by constructing and evaluating logical expressions including non-IP packet attributes of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set referencing said application layer application.

24. [Currently amended] A method for managing and

controlling communication traffic by centralizing access rules in filters including non-IP packet attributes executing within and referencing data available in system kernels, comprising the steps for outbound packet processing from a first node to a second node of:

receiving said packet in the kernel of the operating system of said first node from an application or process at said first node;

processing said packet by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

responsive to said work control block, determining a process or job identifier;

responsive to said process or job identifier, determining job or process attributes; and

executing said filters by constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set.

25. [Currently amended] The method of claim 24, further comprising the steps for inbound packet processing from said second node to said first node of:

initially operating said kernel at said first node to determine a target application for said packet at said first node by executing a look-ahead function within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;.

26. [Canceled]

27. [Canceled]

28. [Canceled]

29. [Currently amended] A method for managing and controlling communication traffic by centralizing the access rules, comprising the steps for outbound packet processing from a first node to a second node of:

receiving said packet in the kernel of the operating system of said first node from an application or process at said first node, said kernel including a filter processor for constructing and evaluating logical expressions including non-IP packet attributes of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set;

processing said packet within the IP layer of a TCP/IP stack;

by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

determining a user ID control block from said work control block;

from the user ID control block determining attributes for said user; and

passing said attributes to said filter processor for managing and controlling communication traffic.

30. [Currently amended] The method of claim 29, further comprising the steps for inbound packet processing from said second node to said first node of:

initially operating said kernel at said first node to determine a target application for said packet at said first node by executing a look-ahead function within said IP layer of said TCP/IP protocol stack, said TCP/IP protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered.

31. [Canceled]

32. [Canceled]

33. [Canceled]

34. [Currently amended] A method for control and management of communication traffic with respect to a system node, comprising the steps of:

receiving at said system node an inbound packet; and

executing within a protocol stack of the system kernel of said system node a filtering function identifying for said inbound packet a filter including non-IP packet attributes referencing non-packet data, and constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set; and

responsive to said filter, executing a look-ahead function for identifying a target application for said inbound packet; said look-ahead function executed within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered[[;]].

35. [Currently amended] The look-ahead function of the method of claim 34 wherein said protocol stack is a TCP/IP protocol stack, and further comprising the steps of:

passing to a transport layer function identified by an IP header a packet marked ~~non-deliverable~~ deny for determining which user-level process or job is to receive said packet;

receiving from said transport layer an application layer task identifier for said user-level process or job; and thereafter

passing said packet marked by said task identifier to said transport layer for delivery to said application layer task.

36. [Currently amended] System for control and management of communication traffic, comprising:

a system kernel including a filter function and stack data;

said filter function including a filter including non-IP packet attributes selectively referencing said stack data for expressing access rules;

said filter function being responsive to receipt of an outbound packet for determining a source application;

said filter function being responsive to receipt of an inbound packet ~~processing~~ for executing a look-ahead function within the IP layer of a TCP/IP protocol stack to determine a target application; said protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-~~

~~deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and

said filter function being responsive to said source or target application for executing filter processing including constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set.

37. [Currently amended] A system for control and management of aspects of communication traffic within filtering, comprising:

a system kernel;

a protocol stack including an IP layer, a transport layer, a sockets layer, and an application layer for executing within said IP layer of said system kernel, responsive to an inbound IP packet, a look-ahead function by which said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and

filtering code within said system kernel operable with respect to non-IP packet data accessed within said system kernel outside of said protocol stack for controlling and managing said aspects of communication traffic; said filter code for constructing and evaluating logical expressions of arbitrary length including non-IP packet attributes, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set.

38. [Currently amended] A system for centralizing system-wide communication management and control within filter rules including non-IP packet attributes, comprising:

filter statements having a syntax for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values;

said selector referencing data that does not exist in IP packets;

a look-ahead function within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer which, for an inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, for identifying the application layer application to which said packet would have been delivered; and

a filter processor for constructing and evaluating filter statements including logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set.

39. [Currently amended] A system for traversing a portion only of a TCP/IP protocol stack to disallow selective IP packet traffic, comprising:

a system kernel;

a filter processor executing within said system kernel for constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields including non-IP packet attributes, and value set;

said filter processor responsive to an inbound packet for executing within an IP layer a look-ahead function for determining a target application; said look-ahead function operating within said IP layer of said TCP/IP protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;

said filter processor responsive to both inbound and outbound packets for

processing said packet by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

determining a user ID, process or job identifier from said work control block;

from the user ID, process or job identifier selectively determining attributes for said user process or job; and

passing said attributes to said filter processor for managing and controlling communication traffic.

40. [Currently amended] A system for expressing access rules as filters, comprising:

filter statements for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values;

said selector referencing data that does not exist in IP packets for controlling access to an application;

a look-ahead function executing within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for an inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and

a filter processor for constructing and evaluating said filter statements as logical expressions of arbitrary length, each said logical expression selectively including said operator selected from a set of logical operators, alternative filter selector fields including non-IP packet attributes, and value set.

41. [Currently amended] A system for managing and controlling communication traffic by centralizing access rules in filters including non-IP packet attributes executing within and referencing data available in system kernels, comprising:

a computer readable medium;

first code for receiving a packet in the kernel of the operating system of a first node from an application or process at said first node; said kernel responsive to an inbound packet, for executing a look-ahead function within the IP layer of a TCP/IP protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;

second code for processing said packet by determining a task ID;

third code responsive to said task ID for determining a corresponding work control block;

fourth code responsive to said work control block for determining a process or job identifier;

fifth code responsive to said process or job identifier for determining job or process attributes;

sixth code for executing said filters by constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set; and wherein

said first, second, third, fourth, fifth, and sixth code is recorded on said computer readable medium.

42. [Canceled]

43. [Currently amended] A system for control and

management of communication traffic with respect to a system node, comprising:

a filtering function executing within the IP layer of a protocol stack of the system kernel of said system node identifying for an inbound packet a filter referencing non-packet data; and

a look-ahead function responsive to said filter including non-IP packet attributes for identifying a target application for said inbound packet; said look-ahead function functioning within said IP layer of [[a]] said protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;; and

a filter processor for constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set.

44. [Canceled]

45. [Currently amended] A computer program product for control and management of aspects of communication traffic within filtering, said computer program product comprising:

a computer readable medium;

first program instructions to receive IP packet data into a TCP/IP protocol stack executing within a system kernel including, for processing an inbound IP packet, a look-ahead function within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;

second program instructions to execute filtering code within said system kernel with respect to non-IP packet data accessed within said system kernel outside of said TCP/IP protocol stack by constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set; and wherein

said first and second program instructions are recorded on said medium.

46. [Currently amended] A computer program product for centralizing system-wide communication management and control within filter rules, said computer program product comprising:

a computer readable medium;

first program instructions to execute filter statements including non-IP packet attributes having a syntax for accepting parameters in the form of a selector, each selector specifying selector field, a logical operator selected from a set of a plurality of logical operators, and a set of values; and

second program instructions to cause said selector to reference data that does not exist in IP packets, said data including application layer indicia obtained for an incoming packet by a look-ahead function; said look-ahead function executing within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and wherein

said first and second program instructions are recorded on said medium.

47. [Currently amended] A computer program product for managing and controlling communication traffic by

centralizing access rules in filters including non-IP packet attributes executing within and referencing data available in system kernels, said computer program product comprising:

a computer readable medium;

first program instructions to receive said packet in the kernel of the operating system of said first node from a process at said first node;

second program instructions to process said packet by determining a task ID;

third program instructions, responsive to said task ID, to determine a corresponding work control block;

fourth program instructions, responsive to said work control block, to determine a process or job identifier;

fifth program instructions, responsive to said process or job identifier, to determine job or process attributes; and sixth program instructions to execute a filter processor for constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields including non-IP packet attributes, and value set; and wherein

said first, second, third, fourth, fifth, and sixth program instructions are recorded on said medium.

48. [Currently amended] The computer program product of claim 47, wherein said protocol stack is a TCP/IP protocol stack, and said computer program product further comprising for inbound packet processing from said second node to said first node:

sixth program instructions to initially operate said kernel at said first node to determine a target application for said packet at said first node by executing a look-ahead function within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;; and wherein

said sixth program instructions are recorded on said medium.

49. [Currently amended] A computer program product for control and management of communication traffic, comprising:

a computer readable medium;

first program instructions for expressing access rules as filters including non-IP packet attributes referencing system kernel data;

second program instructions, for outbound processing, for determining a source application;

third program instructions, for inbound packet processing, for executing a look-ahead function to determine a target application; said look-ahead function operating within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;

fourth program instructions, selectively responsive to said source and target application, for executing filter processing including constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set; and wherein

said first, second, third, and fourth program instructions are recorded on said computer readable medium.

50. [Currently amended] A computer program product for

control and management of aspects of communication traffic within filtering, comprising:

a computer readable medium;

first program instructions for receiving IP packet data into a TCP/IP protocol stack including an IP layer executing within a system kernel;

second program instructions for executing filtering code within said IP layer of said system kernel with respect to non-IP packet data accessed within said system kernel outside of said TCP/IP protocol stack; said filtering code constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields including non-IP packet attributes, and value set; and wherein

said first and second program instructions are recorded on said computer readable medium.

51. [Currently amended] A computer program element for centralizing system-wide communication management and

control within filter rules, comprising:

a computer readable medium;

first program instructions for providing filter statements syntax for accepting parameters in the form of a selector, each selector specifying selector field, a logical operator, and a set of values,

second program instructions for executing filtering by constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including said logical operator selected from a set of logical operators, at least one said selector field including non-IP packet attributes, and at least one said value;

said selector referencing data that does not exist in IP packets including data obtained, for an inbound IP packet, by executing a look-ahead function within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and wherein

said first and second program instructions are recorded on said computer readable medium.

52. [Currently amended] A computer program product for managing and controlling communication traffic by centralizing access rules in filters on non-IP packet attributes executing within, and referencing data available in, system kernels, comprising:

a computer readable medium;

first program instructions for receiving said packet in the kernel of the operating system of said first node from an application or process at said first node;

second program instructions for processing said packet by determining a task ID;

third program instructions, responsive to said task ID, for determining a corresponding work control block;

fourth program instructions, responsive to said work control block, for determining a process or job identifier;

fifth program instructions, responsive to said process or job identifier, for determining job or process attributes;

sixth program instructions for executing a filter processor for constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set; and wherein

said first, second, third, fourth, fifth, and sixth program instructions are recorded on said computer readable medium.

53. [Currently amended] The computer program product of claim 52, further comprising for inbound packet processing

from said second node to said first node:

seventh program instructions initially operating said kernel at said first node to determine a target application for said packet at said first node by executing a look-ahead function within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and wherein

said seventh program instructions are recorded on said computer readable medium.